API MAPS AND CURL

Requirements

1. Curl must be installed on the machine.

Info

1. These commands can either be sent in the JSON format or X-WWW-FORM-URLENCODED content-type.
2. In this document the curl command examples in Table 2 use X-WWW-FORM-URLENCODED content-type.

API Map and CURL

Table 2 below shows a numbered list of curl commands and an example of how they should be implemented. The numbering in Table 1 gives a description of the corresponding numbered row in Table 2. Table 1 shows the API Map and Table 2, shows the Curl example implementations.

Table 1: API function mapping

Function	API / Description
1. Create a new user	**/createuser/** - The API takes a username, password and email and then register the user. When the user is successfully registered, a JSON response with the registration details is sent back.
2. Login	**/login/ -** The API takes a username and password, authenticate them against the created user. If the user credentials are correct then a JSON response with an Authentication Token is returned. This token must be passed into the header of all

	following requests for authorization. The header must look like this : "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c"
3. Create ,update and delete a savings account	**/savings/ -** The API's action depends on the http verb, POST: A savings account is created and an initial **balance** >=50 must be passed on the body. The response returns the **acc_id** in a response JSON. PATCH: An existing savings account (specified in the URL parameter by **?acc_id=<int>**) is edited. Specifically, the amount is manually altered. This API also takes a **balance** DETELE**:** An existing savings account (specified in the URL parameter by **?acc_id=<int>**) is deleted. **NB: The "acc_id" for the user's accounts can be found by making a request on /balances/ (see Table 1, number 7)**
4. Create ,update and delete a savings account	**/credit/ -** The API's action depends on the http verb, POST: A credit account is created and an initial **balance** >=-20000 must be passed on the body. The response returns the **acc_id** in a response JSON. PATCH: An existing credit account (specified in the URL parameter by **?acc_id=<int>**) is edited. Specifically, the amount is manually altered. This API also takes a **balance** DETELE**:** An existing credit account (specified in the URL parameter by **?acc_id=<int>**) is deleted. **NB: The "acc_id" for the user's accounts can be found by making a request on /balances/ (see Table 1, number 7)**
5. Transact in a savings account	**/savings/transact/** – only the POST method is allowed. The action (withdrawal or deposit) is specified by the URL parameter **?type=< widthdraw | deposit >** The body of the request should specify the "amount" and "acc_id" to perform the transaction on. **NB: The "acc_id" for the user's accounts can be found by making a request on /balances/ (see Table 1, number 7)**
6. Transact in a credit account	**/credit/transact/** – only the POST method is allowed. The action (withdrawal or deposit) is specified by the URL parameter **?type=< widthdraw | deposit >**

	The body of the request should specify the “amount” and “acc_id” to perform the transaction on. **NB: The “acc_id” for the user’s accounts can be found by making a request on /balances/ (see Table 1, number 7)**
7. Get accounts, balances, last 10 transactions	**/balances/ -** This API takes a GET request and doesn’t need any data to be passed in the request body. It will return a JSON response with the values as described by requirement 7.
8. Get CSV file	**/csv/ -** This API generates a CVS file and then returns it. If entered in the web browser, the CSV file will download.

NB: Where there is Authorization, the token used must be the one returned by the Login request

Table 2: List of functions and relating curl implementation examples

	URL params	**Body values**	**Curl command examples**
1. Create a new user		Username, password, email	curl -X POST http://localhost:8000/createuser/ -H "Content-Type: application/x-www-form-urlencoded" –d "username=katlego3" -d "password=katlego2" -d "email=sk@gmail.com"
2.Login		Username, password	curl -X POST http://localhost:8000/login/ -H "Content-Type: application/x-www-form-urlencoded" -d "username=katlego2" -d "password=katlego2"

3. Create ,update and delete a savings account	acc_id	balance	**CREATE:**
			curl -X POST http://localhost:8000/savings/ -H "Content-Type: application/x-www-form-urlencoded" -H "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c" -d "balance=50"
			UPDATE:

			curl -X PATCH http://localhost:8000/savings**/?acc_id=4** -H "Content-Type: application/x-www-form-urlencoded" -H "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c" -d "balance=500" **DELETE:** curl -X DELETE http://localhost:8000/savings**/?acc_id=2** -H "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c"
4. Create ,update and delete a savings account	acc_id	balance	**UPDATE:** curl -X POST http://localhost:8000/credit/ -H "Content-Type: application/x-www-form-urlencoded" -H "Authorization: Token d9d81961f0d574c76fb5052e259027cae5e1f140" -d "balance=0" **UPDATE:** curl -X PATCH http://localhost:8000/credit**/?acc_id=2** -H "Content-Type: application/x-www-form-urlencoded" -H "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c" -d "balance=5000" **DELETE**: curl -X DELETE http://localhost:8000/credit**/?acc_id=2** -H "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c"
5. Transact in a savings account	Type (**withdraw or deposit**)	Acc_id, amount	**DEPOSIT:** curl -X POST http://localhost:8000/savings/transact**/?type=deposit** -H "Content-Type: application/x-www-form-urlencoded" -H "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c" -d "amount=50" -d "acc_id=3" **WIDTHDRAW:**

			curl -X POST http://localhost:8000/savings/transact**/?type=widthdraw** -H "Content-Type: application/x-www-form-urlencoded" -H "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c" -d "amount=50" -d "acc_id=3"
6. Transact in a credit acccount	Type (**withdraw or deposit**)	Acc_id, amount	**DEPOSIT:** curl -X POST http://localhost:8000/credit/transact/**?type=deposit** -H "Content-Type: application/x-www-form-urlencoded" -H "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c" -d "amount=50" -d "acc_id=3" **WIDTHDRAW:** curl -X POST http://localhost:8000/credit/transact**/?type=widthdraw** -H "Content-Type: application/x-www-form-urlencoded" -H "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c" -d "amount=50" -d "acc_id=3"
7. Get accounts, balances, last 10 transactions			curl -X GET http://localhost:8000/balances/ -H "Accept: application/json" -H "Authorization: Token 1a5a0d136d6cdaa0fd6f4796d5db39ca54ea323c"
8. Get CSV file			**View from terminal** curl -X GET http://localhost:8000/csv/ **Download from browser** http://localhost:8000/csv/